Dev/Building

From RibTools

Contents

The Code

RibTools is currently released under the LGPL license (http://www.gnu.org/copyleft/lesser.html) . The code is open for anyone that wants to use it, as long as it's properly redistributed.

RibTools currently compiles only for Windows and using Visual Studio 2008. The code is written with portability in mind and support for Linux builds is planned for the future.

- Get the source code (http://ribtools.googlecode.com/)

Directory Structure

As obtained from the Subversion repository:

DImage/	Bitmap images library
DispDrivers/	Display drivers sources
DistribSrc/	Source assets
DMath/	Math library sources and project
docs/	Documentation related files (currently just some images)
DSystem/	Base system library sources and projects
external/	External libraries, directly included
externals/	External libraries included automatically by SVN
Extra/	**(unused)**
linux/	Linux main project file **(unused)**

osx/	OS X main project file **(unused)**
pc/	Windows main project file
RI_System/	Core rendering engine and RenderMan-interface
RibRender/	RibRender application project and sources
RibRenderLib/	A library that groups most functionalities used by both *RibRender* and *RibRenderToy*
RibRenderServer/	RibRenderServer application project and sources
RibRenderToy/	RibRenderToy application project and sources
RSLCompilerCmd/	RSLCompilerCmd application project and sources
RSLCompilerLib/	RSL compiler library
build.bat	Script that builds the **assets** from **DistribSrc/** into **Distrib/**
license.txt	License file
make_install.bat	Script that builds a .zip file for binary distribution (assumes EXEs have been manually compiled with VS)
readme.html	Read-me file

An additional directory named **Distrib/** is generated as part of the build process. This directory is automatically generated, and it can therefore be safely erased (unless one has any important changes in it !)

The **Distrib/** directory contains both data coming from **DistribSrc/** and executables generated when compiling with Visual Studio (see below).

There are some directory naming conventions used through the project:

- **pc/** directories hold files specific to Windows builds (usually Visual Studio solution and project files)
- **linux/** directories hold files specific to Linux builds
- **osx/** directories hold files specific to Mac OS X builds
- **src/** directories hold *.cpp* files
- **include/** directories hold *.h* files

Compiling and running from VS

Unpack libtiff

libtiff's source code is kept as the *tiff-3.9.2.zip* archive under the **externals/** folder. Unzip the *tiff-3.9.2.zip* in its own location.

As the result of unzipping, the following directory structure is expected:

ribtools

externals/
tiff-3.9.2/

build/
libtiff/
COPYRIGHT
[...]

Building the solution

1. Double click on ***build.bat*** to run a script that will create the directory **Distrib/**
2. Open the directory **pc/** from the main directory

1. Double click on ***RibTools.sln*** inside **pc/** to launch the main solution with Visual Studio 2008
2. From the Standard Toolbar select a *solution configuration* (**Debug** or **Release**) and a *platform configuration* (**Win32** or **x64**)

1. Build the solution by selecting the option from the **Build** menu (or press F7 or Ctrl+Alt+B, depending on the VS configuration)

Once the build is finished the **Distrib/** directory should contain 3 executables:

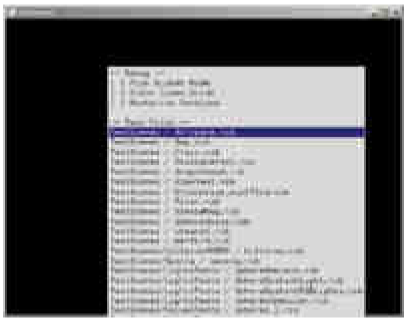
- **Extras/**
- **Install/**
- **TestScenes/**
- **TestsOutput/**
- **Resources/**
- *DelCompiledShaders.bat*
- *MakeTests.bat*
- *readme.html*
- *RibRender.exe*
- *RibRender.pdb*
- *RibRenderServer.exe*
- *RibRenderServer.pdb*
- *RibRenderToy.exe*
- *RibRenderToy.pdb*
- *RSLCompilerCmd.exe*
- *RSLCompilerCmd.pdb*

Note that for every *.exe* file there is a corresponding *.pdb* file.

PDB files hold debug data necessary to display symbols when debugging with Visual Studio's debugger.

A little test

- From the **Distrib/** directory, launch ***RibRenderToy.exe***
- Right-click inside the RibRenderToy window and select a test scene from the pop-up menu (*Airplane.rib* is a safe bet)



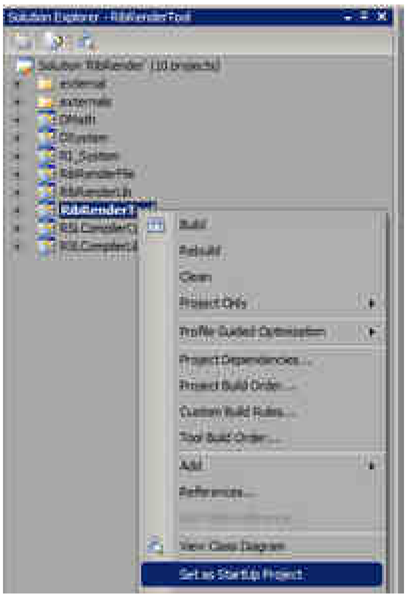
- After a little while the scene should appear in all it's beauty (!)

Running from Visual Studio

In order to run form visual studio, a one time setup is necessary.

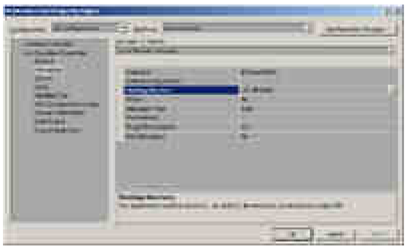
Set the choosen application as the StartUp project

Right-click on an executable project (e.g. RibRenderToy), and choose **Set as StartUp Project** from the pop-up menu.



Setup the working directory

- Right-click on the application project (e.g. RibRenderToy) and select **Properties** from the pop-up menu
- Find the **Working Directory** setting in the **Properties** dialog by selecting *Configuration Properties* -> *Debugging*
- Set the **Working Directory** to **..\..\Distrib**



Note: Set this for all the configurations (*Release* and *Debug*) and all platforms (*Win32* and *x64*). A common mistake is to set up the Working Directory only for the current configuration and forget about the other configurations.

Run (!!!)

Run the application with or without the debugger by pressing F5, Ctrl+F5 (or whatever is your configuration 8)

Retrieved from "http://ribtools.com/wiki/Dev/Building"

- This page was last modified on 26 January 2010, at 12:32.